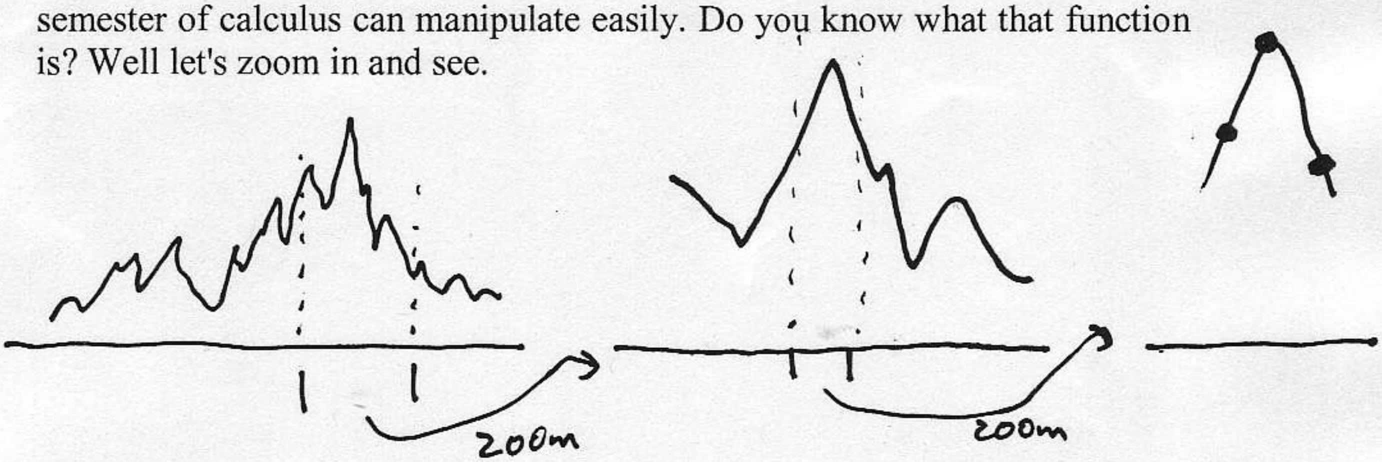
Optimization and You: Preparation for Inquiry 3

Time Out: An interesting thought as we start to learn about optimization and curve fitting. Look at the graphs below that describe scientific data. Do they have much in common? Well one feature they share is a maximum and minimum value. These would be optimum values that we would want to find in a response surface, but no great surprise. Here is something else that is more surprising. No matter how complicated each of these functions may be, no matter how different they are from one another, the same mathematical function describes the peak shape at the maximum or minimum. Not only that, it is a function that everyone with a single semester of calculus can manipulate easily. Do you know what that function is? Well let's zoom in and see.

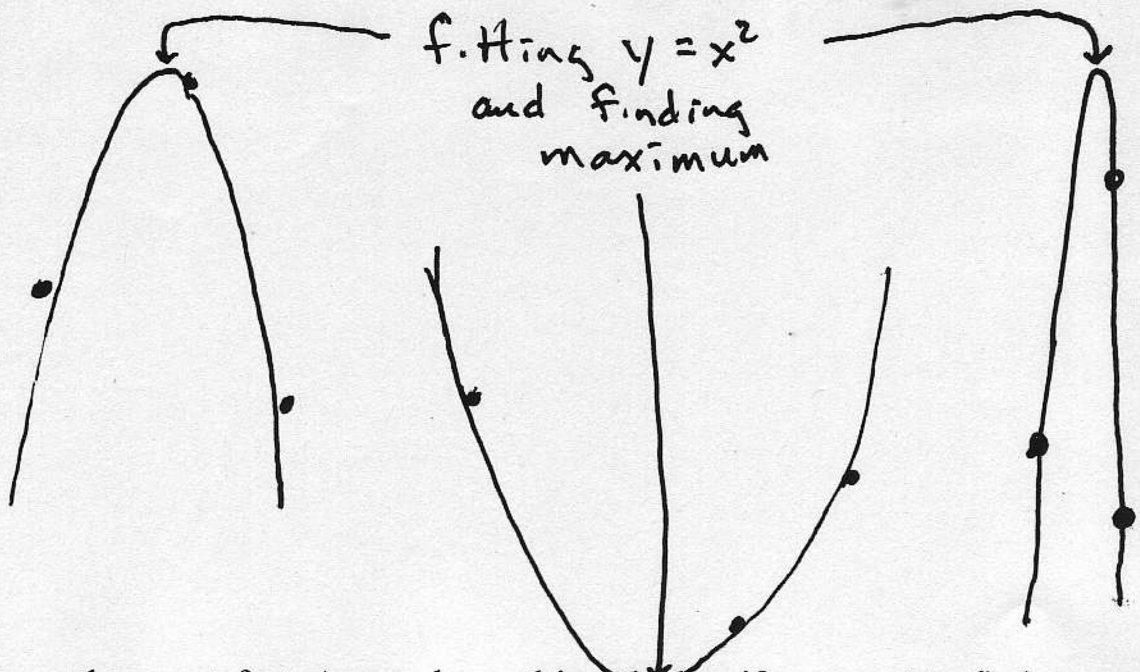


Notice that no matter what happens away from the function maximum, as we get closer and closer to the top, we start to see shapes that are increasingly similar. In every case the function is observed to increase, hit a maximum, and then goes back down.

So what function describes each of these peak maxima and minima? A simple parabola:

$$y = k x^2$$

It doesn't matter how complex the function is that describes the overall data, when you approach the limit of a peak maximum or minimum, you can use a simple parabolic function to model the data you acquire. Three data points describe any local maximum or minimum and can be fit to a parabola to obtain a model function.



Here is another great fact. As you learned in calculus, if you want to find a maximum or minimum, you simply take a derivative of a function and set it equal to zero. Do you know how to take the derivative of a parabola?

$$\frac{d}{dx}x^2 = 2x$$

derivative yields max/min

And if you want to know the area under such a peak, do you know how to find the integral of a parabola?

$$\int x^2 dx = \frac{x^3}{3} \leftarrow$$

area under curve

Isn't this great? You might not remember how to take the integral of a hyperbolic cosecant, but on the part of a graph of data that usually provides the greatest amount of information (the maxima or minima), you don't have to know. It is one important reason why so much applied math in the sciences is so darn simple. So much of it can be reduced to simple approximations like this one: the maxima and minima for every function can be modeled by a parabola. After all, scientists don't like having to do complicated math any more than college students.

What you have seen in this time out are two important aspects of the scientific experiment, finding maxima or minima on a response surface, which is called OPTIMIZATION, and then fitting a curve to the data, MODELING DATA.

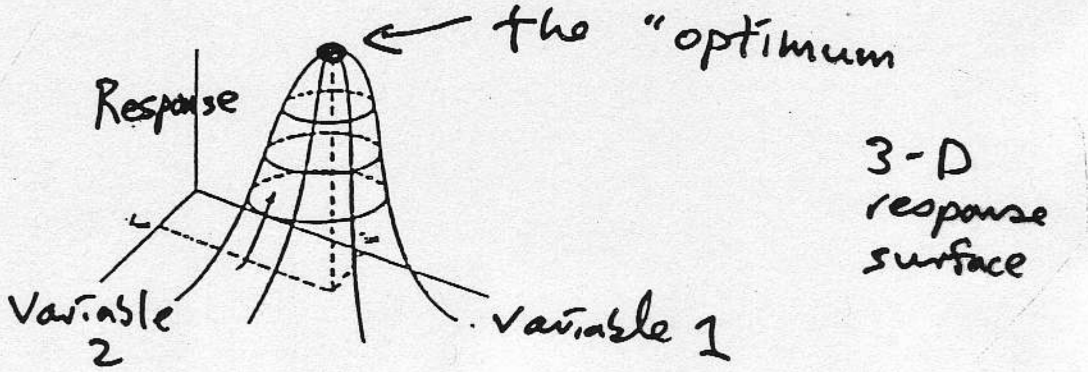
In the next set of notes, the question of how to perform a response surface OPTIMIZATION will be considered. This will be followed by a demonstration that anyone can apply linear least squares to MODEL DATA.

OPTIMIZATION

The new word for the day: OPTIMIZATION. Now optimizing is a desirable trait in everything you do all day long, and that includes your scientific endeavors. No matter what we are doing , we want it to be done the best that it can be done, kind of like the Marines being all they can be. So if I am baking a loaf of bread, I want to make the best tasting bread I can, maybe by MAXIMIZING the taste. In the same way, if I am getting dressed in the morning, I want to MINIMIZE the time it takes to get from my bed to the car. If I am running a chemical reaction in organic lab, I want to get the MAXIMUM yield. If I have a spectrometer, I want to tune it to get the MAXIMUM sensitivity.

So the next question is, how can we most EFFICIENTLY find the optimum? In fact there are all kinds of algorithms for finding the optimum--there is even an entire field of mathematics devoted to it. Even your brain has built into it a collection of algorithms it uses for optimizing how you accomplish things in everyday life. How do you most efficiently and effectively get the MAXIMUM points on an exam? How do you most efficiently convince your date to kiss you as quickly and as often as possible? Do people still date?

What we are going to do today is consider what some of the algorithms for optimizing might be. What you will find is that you could probably come up with them using a little common sense. For example, some of you might be thinking, "gee, I could see how finding an optimum can often be like finding a maximum; for example, maximizing the taste of a loaf of bread by varying temperature and baking time. Well I already know how to find a maximum when I do math, I just take a derivative and set it to zero." Well there you have it, an example of common sense in action. Indeed, the most powerful set of algorithms for finding an optimum are DERIVATIVE methods. Some guy named Newton actually came up with them. The problem is this: when was the last time you took the derivative of baked bread? This brings up a good point. How exactly do we define what it is we are trying to optimize?



The answer to this question requires that we think about RESPONSE SURFACES. A response surface is just that, a surface that results when we vary a bunch of factors and see what kind of results we get. Shown in the figure above is an example of a 3-dimensional response surface. Here, a response (like your exam grade) is shown coming out of the paper. It is obtained by looking at all the different combinations of two factors (like how much sleep you got the night before and how much you studied) to get the RESPONSE (exam grade) that is plotted on a SURFACE. Now understand that for convenience I have shown just a 3-D picture made with two variables and one response. In fact I could have had 28 factors and one response, but I have no idea how to draw a 29 dimensional space. Still, a computer can understand 29 dimensional space and very often, you will deal with more complicated optimizations by using a computer. But for simplicity, we will stick to 3-D optimization (two factors) for this lecture.

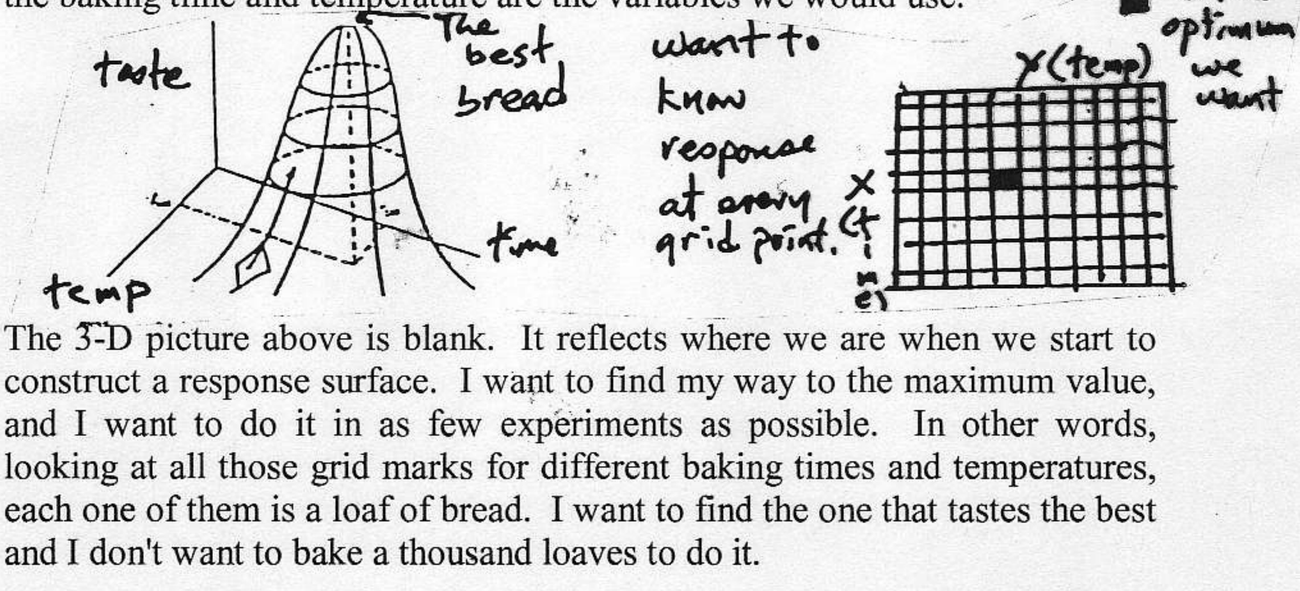
There are an infinite number of examples of three dimensional response surfaces and to prove it, you will get to come up with your very own for the next inquiry. For example, a couple of years ago I gave this assignment and one guy did an optimization in which the response was how long his girlfriend could stand to be around him and the factors were how long he went without showering and how much deodorant he used!

How fast can you optimize? It is one thing to decide to perform an optimization, in other words, to find a maximum or minimum on a response surface. A second question is, how FAST can you find the optimum value?

There are lots of reasons for wanting to be able to efficiently find the optimum. It is far less expensive, you get to leave the lab quicker, and it is a lot more accurate and precise--the response surface is a lot less likely to vary during shorter optimization procedures.

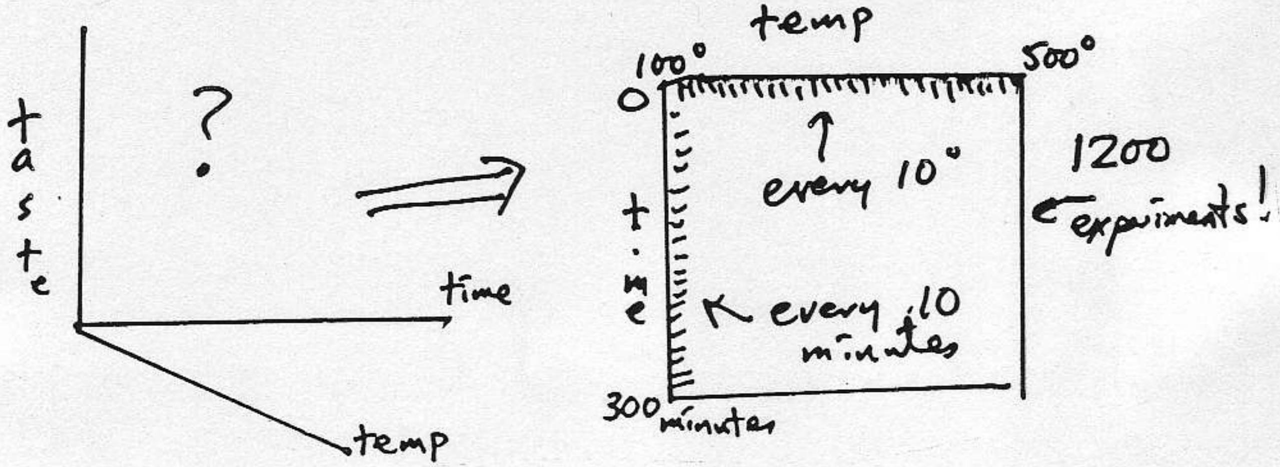
What are the Optimization Methods? The question, then, is what are the ways to find the optimum response? Once again the answers are concepts you could come up with using common sense. The examples we will look at are **"brute force"** which is the slowest and dumbest, but always works; a second is the use of **"trial and error"** in which you rely on prior experience with the response surface; a third method is **derivative** methods which are the fastest, but only work when you can model the response surface with mathematical expressions; and finally, **geometric** methods, which are the way our brains typically work as we plod along the response surfaces of life, trying to reach our goals--not real fast, not real slow, but pretty likely to get us where we want to go.

Our Optimization example: Baking a loaf of good taking bread. The example we will consider as we attempt to optimize is the baking of a loaf of bread. We want a response surface for which good taste is the response, and the baking time and temperature are the variables we would use.



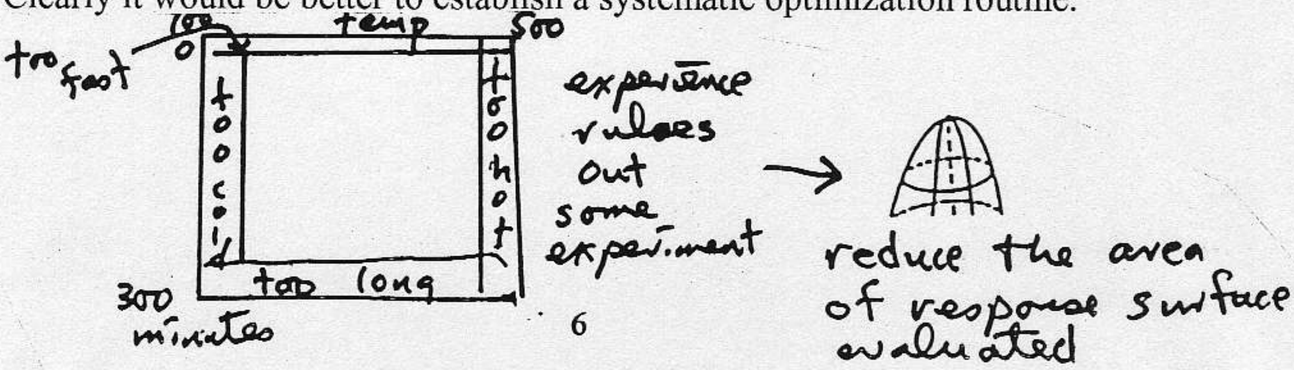
The 3-D picture above is blank. It reflects where we are when we start to construct a response surface. I want to find my way to the maximum value, and I want to do it in as few experiments as possible. In other words, looking at all those grid marks for different baking times and temperatures, each one of them is a loaf of bread. I want to find the one that tastes the best and I don't want to bake a thousand loaves to do it.

METHOD 1. BRUTE FORCE. The most obvious, and clearly worst way to optimize, is brute force--we collect a response for every possible combination of factors. The number of experiments required would be determined by the number of factors, the reasonable boundaries we apply to each variable and the resolution of the measurement. For example, we bake some bread and want to optimize the temperature and the amount of time. We decide to try every temperature between 100 and 500 degrees, with a 10 degree increment, and 0 minutes to 300 minutes, with a 10 minute interval. We end up making a matrix with 40 gradations in the temperaure, and for each of these, 30 gradations of time. With and average baking time of 150



minutes, and with 40 x 30 = 1200 bread bakings, we are talking about 3.5 years to collect all the data in a single bread maker. On the bright side, brute force takes absolutely no mental effort. And, admit it, we use brute force all the time when we don't want to have to think. (We often combine laziness with the brute force method, and stop before we can have fully mapped the surface. This falls in the "take a wild guess" category of optimization. Much quicker then brute force, though rarely right.)

METHOD 2. COMMON SENSE AND EXPERIENCE (TRIAL AND ERROR). Often we empioy experience and common sense in a kind of optimization (trial and error) which is far more satisfactory than brute force. We can cut down on those 1200 bread baking experiments because we know that bread is unlikely to bake very much when it has been in the oven for 0 minutes. So you know you don't need to test those places on the response surface. This approach to optimization is the one we most often employ when we have prior knowledge of a response surface. The problem is, very often in science we are working with totally unknown processes. Clearly it would be better to establish a systematic optimization routine.



METHOD 3. DERIVATIVE METHODS. Think back to calculus. What was the one thing you leaned that somehow seemed to be relevant, seemed to make sense? Here it is,

"How do you find a maximum?"

TAKE THE DERIVATIVE AND SET IT TO ZERO.

And what are you doing when you find a function maximum? You are finding an optimum location on a response function.

What this observation from calculus suggests, then, is that the best approach to efficient optimization would be to find a response function for your response surface, and take its derivative. So how do you find a response function? One way is to collect enough data on the response surface to allow you to fit a function to the data. Then you can take a derivative of the function and set it to zero. The other approach is to come up with a model function for the response surface from theory. Then, without even going into the lab, you can perform a mathematical operation to find the optimum value.

Let's look at an example, and again, let's bake some bread. Now granted response surfaces for bread making aren't exactly easy to come by. But just for kicks, lets assume it has the following shape to describe the response surface:

$$R = \underset{\text{baking time (hours)}}{e^{-(b-1)2}} + \underset{\text{temperature (hundreds of degrees)}}{e^{-(t-2)2}}$$

then

$$0 = \left.\frac{\partial R}{\partial b} + \frac{\partial R}{\partial t}\right\} \text{this is the derivative of the response. Set it to } \emptyset$$

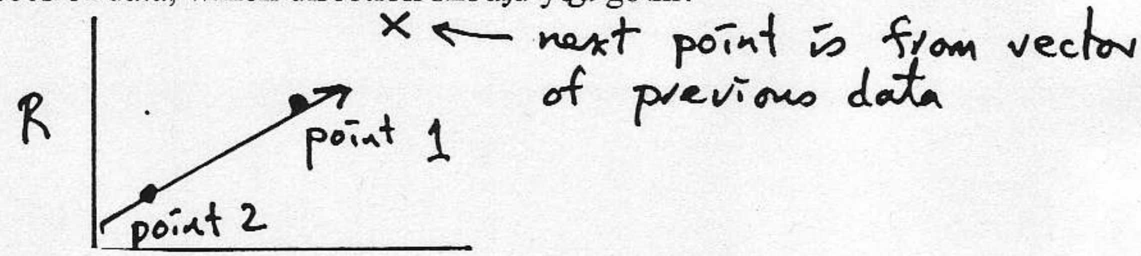
and by inspection, when baking time$_{opt}$ = 60 minutes and temperature$_{opt}$ = 200°, the optimum response is the value R_{max} = 2.0. (A completely arbitrary number representing the best taking bread.)

Again, though, the problem is that far too often for scientists, there is no Einstein sitting around coming up with an equation to describe the response surface. As a consequence, using derivative methods is of little value to experimentalists. We will have to find something more practical.

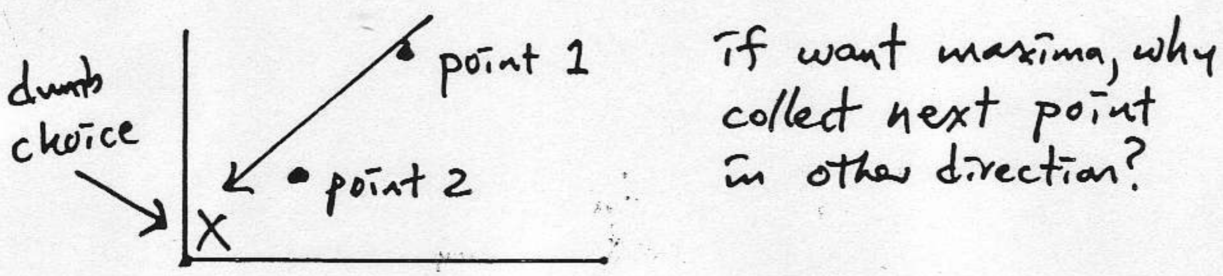
An aside: although derivative methods are letting us down here, next week when we look at how to fit curves to data, we will derive linear least squares. And guess what, we will be using the derivative method to optimize the curve fitting process!!

METHOD 4. SIMPLEX OPTIMIZATION.

Suppose you are on an unknown response surface and you have to make a common sense decision about what piece of data to collect after collecting your first data. For example, on this surface, after collecting these two pieces of data, which direction should you go in?



Clearly, if you want to move in the direction of the maximum response, you point a vector up a hill, not down.



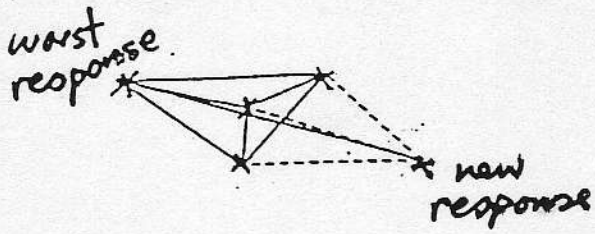
This is the basis of a kind of real time optimization from a class of geometric optimizations. Geometric optimizations are of a kind that you would use most commonly in an environment where you are collecting and assessing data as you determine the next data point to acquire on the response surface.

For our purposes here, we will look at a non-derivative N-dimensional geometry-based algorithm called the SIMPLEX. Experimental scientists love the simplex, because even they can understand the math. It is actually an okay algorithm, but it is pretty unsophisticated. There are plenty of others that are far faster and more accurate in converging to an optimum. But lets stick with the simplex.

A simplex is a geometrical figure in N space with N+1 vertices. Each vertex is a combination of variables and represents a vector in N space. In the simplest simplex algorithm we go from some worse location on the response surface to some better location by reflecting a vector through the middle of the simplex to some equal and opposite distance from the worst vertex in the original simplex.

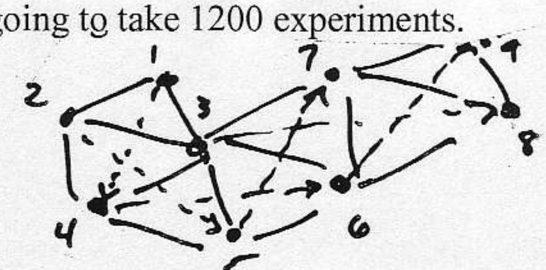
Now rather than use the simple two-point optimization shown above, it is better to use more data in drawing the vector. For example, in the figure below is a 4-D simplex (four vertices). A reflection is made through the old (solid line) pyramid-shaped simplex to form a new (dashed line) simplex. The common sense idea is that if you move away form the worst point in a collection of vertices, you must be headed in the right direction. In general, in a well-behaved response surface, this is a good assumption.

a 4-D simplex, 3 factors, 1 response



x = vertex corresponding to experiment data point

Well the idea of the algorithm is to make a new and improved simplex over and over again, until we don't get any higher on the response surface. To a scientist, this means we do a collect a new data point (bake a new loaf of bread), get a new response, and form a new simplex, throwing out the old worst point. Supposedly far fewer bread baking experiments would have to be done using a simplex. This is likely, sense our brute force method was going to take 1200 experiments.



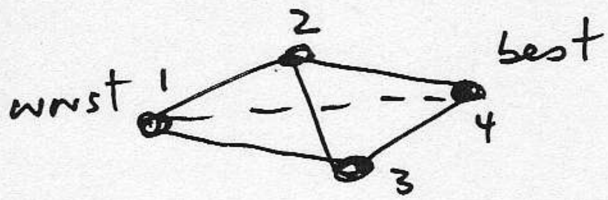
here 9 data points are acquired as we form one simplex after another moving to the best response

Now, just exactly what is the math for a simplex? Believe it or not, you've had a course in it, way back in vector algebra. You remember, it kind of had to do with making calculations of distances between points in space. Well basically that is what we will do here. To perform the simplex in 3-D space we make a shape that look like a pyramid. To make a simplex in 2-D space we make a shape that looks like a triangle. The latter sounds easier to draw so that is what we will do here. It is actually the one you could use to march up a three dimensional surface with two variables and a response.

Step by step simplex optimization process.

Step 1. Run three separate bread baking experiments and form your first simplex.

Step 2. Run a fourth bread baking experiment at a location that is reflected through the first simplex away from worst response vertex.



collect 1, 2, 3 and decide which is worst, Then draw vector to 4.

Now mathematically, this value is determined with a bit of matrix algebra by a two step process:

i) calculate P = $\frac{1}{k-1} \sum V_j$

where k is the size of the simplex (3 here) and V_j are the vertices through which the reflection is drawn.

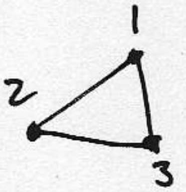
ii) calculate $V'_i = 2P - V_i$

V'_i replaces V_i, the worst (lowest R) vertex in the simplex

Repeat Step 2 until R_{opt}, the maximum on the response surface is reached. This is assumed to occur if R_{opt} stops improving.

EXAMPLE OF A SIMPLEX

Step 1. Select three sets of conditions for temperature and baking time as good starting guesses. Then collect R_{opt} values from the three experiments. Three vertices can now be drawn for our first simplex.

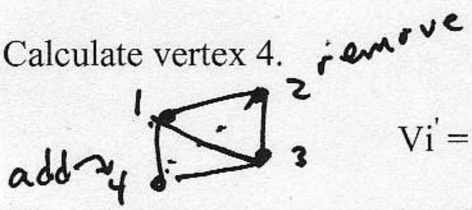


Simplex 1

Vertex	b	t	R_{opt}	
1	0.4	0.5	0.8	← worst case, throw this out
2	0.7	0.7	1.10	
3	0.3	0.8	0.85	

Step 2. Eliminate vertex 1 which has the lowest R_{opt}.

Calculate vertex 4. $\qquad P=1/2((0.7,0.7) + (0.3, 0.8)) = (0.5, 0.7)$

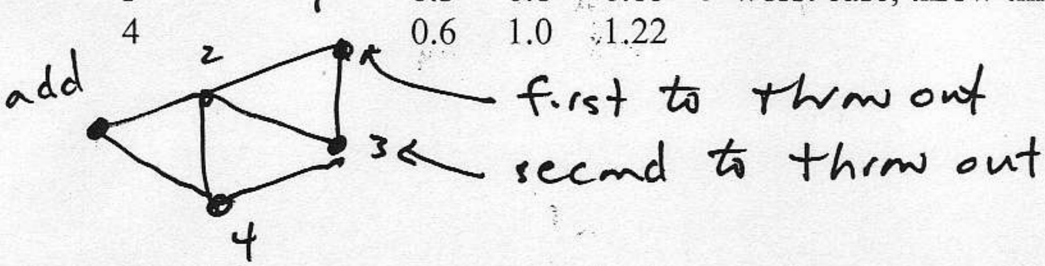


$Vi' = 2(0.5,0.7) - (0.4, 0.5) = (0.6,1.0)$

We now have a new vertex for our second simplex which is supposedly better than vertex 1 which we threw out.

Simplex 2

Vertex	b	t	R_{opt}	
2	0.7	0.7	1.10	
3	0.3	0.8	0.85	← worst case, throw this out
4	0.6	1.0	1.22	



Step 3. Now we construct a new simplex by replacing vertex 3 with vertex 5.

Calculate vertex 5: $\qquad P = (0.7,0.7) + (0.6, 1.0) = (0.65, 0.85)$

$$V_5 = 2(0.65,0.85) - (0.3, 0.8) = (1.0,0.9)$$

And we construct a new simplex with vertices 2, 4, 5

Simplex 3

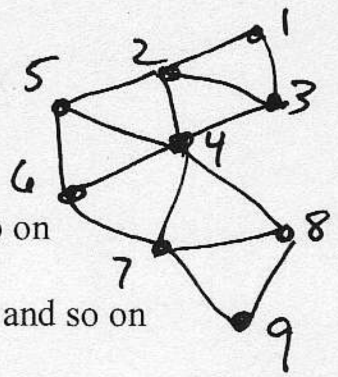
Vertex		b	t	R_{opt}
2	0.7	0.7	1.10	
4	0.6	1.0	1.22	
5	1.0	0.9	1.30	

and so on

and so on

and so on

and so on



One of these days we hope to approach the optimum R value which we happen to know is 2.0 because we had Einstein do a thought experiment. Why don't you just continue the simplex to completion and determine just how many bread baking experiments you avoid by performing a simplex instead of using the brute force approach. You will find that it takes about 14 experiments to get to the optimum, a little better than the 1200 using the brute force approach.